



ALICE Analysis Models

Anton Alkin (CERN) on behalf of ALICE Collaboration

O² DPL Analysis Framework¹

FairMQ transport layer is a message passing toolkit defining core block of architecture and inter-process communication in a medium-agnostic way

Data Model is a programming-language-agnostic extensible description of messages being exchanged, allowing efficient transport and mapping to data objects in memory

Data Processing Layer (DPL) hides the complexity of the abstract transport layer and low-level data model, provides concise logical description of a data flow

Data taking and reconstruction

Timeframes: data comes in batches (~ 10) requiring SoA approach

Calibration: serializable C++ and ROOT objects stored in an online database

Storage: ROOT trees

Analysis

Calibration: same as reconstruction, mostly ROOT objects (histograms)

Input: ROOT trees unpacked into Apache Arrow tables

Output: predominantly ROOT objects (histograms) and trees

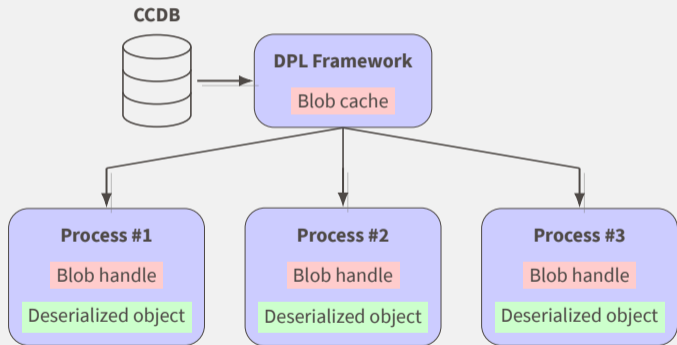
¹A. Alkin et al. (ALICE), EPJ Web Conf. **251**, 03063 (2021).

C++ objects serialization

- ROOT serialization is used to deliver generic C++ and ROOT objects to consumers
- Cached binary blobs are managed by the framework and sent^a as messages to the processes in the workflow
- De-serialization is done by the individual process that maintains its own copy of the object

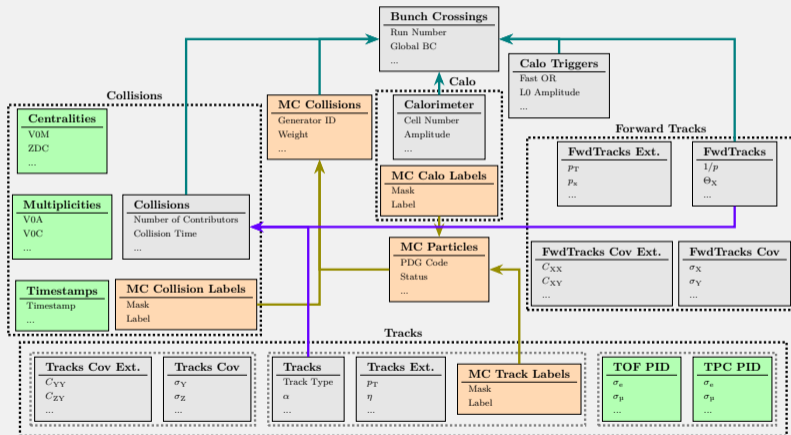
^aor used directly with shared memory backend

Constants and Calibration Database (CCDB)



Data persistence

Abstract Data Model



On-disk

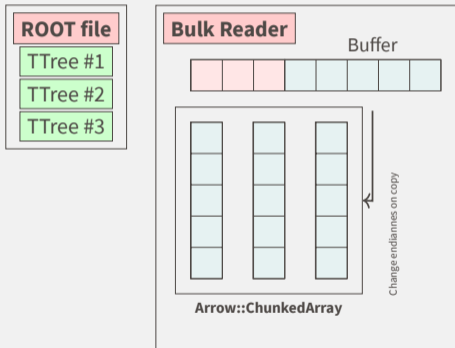
ROOT Files

/Users/aalkin/O2/scratch/LHC21

DF_2595827105001;1

- O2ambiguousfwdtr;1
- O2ambiguousmfttr;1
- O2ambiguoustrack;1
- O2bc;1
- O2calo;1
- O2calotrig;1
- O2cascade_001;1
- O2collision;1
- O2fdd_001;1
- O2ft0;1
- O2fv0a;1
- O2fwdtrack;1
- O2fwdtrackcov;1
- O2mccollision;1
- O2mccollisionlabel;1
- O2mcfwdtracklabel;1
- O2mcmfttracklabel;1
- O2mcparticle_001;1
- O2mctracklabel;1
- O2mfttrack;1
- O2origin;1
- O2track;1
- O2trackcov;1
- O2trackextra;1
- O2v0_001;1
- O2zdc;1

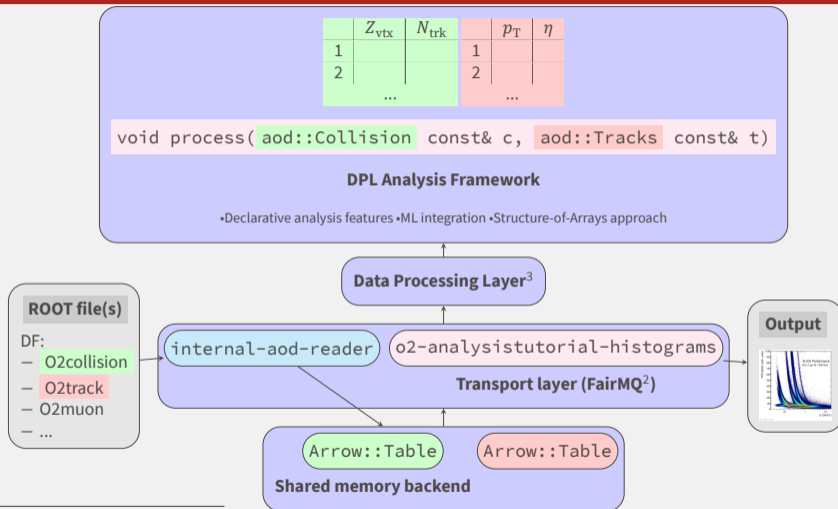
Bulk reading



- Unpacking happens “in place”, data needs to be copied out of the buffer
- Endiannes needs to be changed if using `GetBulkRead().GetEntriesSerialized()`^a
- Data can be read directly in Arrow-provided or shared memory pool
- It is possible to create contiguous array upfront or create a chunked array and use Arrow’s methods for data consolidation

^aRequired as a workaround for reading variable-size array branches

End-user analysis



²M. Al-Turany et al., J. Phys. Conf. Ser. **513**, edited by D. L. Groep and D. Bonacorsi, 022001 (2014).

³G. Eulisse et al., EPJ Web Conf. **214**, edited by A. Forti et al., 05010 (2019).





Efficient container for analysis output objects

- Previous analysis framework relied on `TList`s for output
- `FindObject()` is (was) slow when accessing, e.g. histograms to fill in 2- and 3-track loops
- Using fixed size hash-map gives non-negligible advantage for nested loops





Per-lookup Benchmark	Time ms	CPU Time ms	Comment
HR/4	1.74	1.65	$d = 0$
HR/128	2.00	1.98	$d = 0.04$
TList/4	2.16	2.15	
TList/128	2.51	2.49	

<https://github.com/AliceO2Group/AliceO2/blob/dev/Framework/Core/src/HistogramRegistry.cxx>

RDataFrame

-  Can be directly created from Arrow
-  Declarative expressions need to be recompiled for each timeframe
-  Requires significant rewrite of legacy analysis code
-  Most of the functionality, including declarative features is already implemented in O² framework data model

RNTuple

-  Strongly typed, no need to change endianness on read
-  Direct mapping to memory
-  Potentially a much better match to abstract data model and Arrow memory layout
-  Old files with TTrees need to be converted


ONNX

- Currently integrated in O² framework
- Inference is fully supported, pre-trained models will be provided through CCDB
- Helper API is being developed


TMVA SOFIE

- Considerable performance boost for inference
- Compiled models limit flexibility of the framework
- No integration needed, ROOT libraries are directly usable at analysis level

JIT compilation

- **Gandiva**: LLVM-based engine for bulk operations on Arrow tables
- Supported in O² framework through parsing C++ expressions
- DSL includes arithmetics, math functions and conditional expressions, operating on columns
-  ROOT includes its own LLVM of different version necessitating build system workarounds

C++20

- Moving to C++20 from current C++17 will simplify most of the templated code
-  Migration pending due to various dependencies including ROOT

- ROOT is an integral part of ALICE analysis infrastructure and will remain such in the foreseeable future
- Support for new C++ standards is very welcome, as C++ remains the crucial component for high-performance applications
- Developments such as RNTuple provide significant boost
- Better interoperability with Apache Arrow is desired, as it is currently a very popular way to deliver and process columnar data

BACKUP

Data flow

